



**Universidad Nacional Autónoma
de México**



Facultad de Ingeniería

Asignatura:
Fundamentos Técnicos de VFX

Profesor:
Ing. Carlos Aldair Roman Balbuena

Proyecto Final:
**Desarrollo de un prototipo de un Videojuego:
“Es Lime Al Ataque”.**

No. de cuenta Alumno:
318352969.

Grupo: 3

Semestre: 2026-1

Fecha: 21 noviembre de 2025

Información del Proyecto.

Nombre del proyecto: Es Lime Al Ataque

Fecha del proyecto: 21 de noviembre de 2025

Tipo de desarrollo: Videojuego en plataforma PC.

Administrador del proyecto: 318352969

Objetivos.

- Desarrollar un prototipo de un medio de entretenimiento; siendo este un videojuego 3D aplicando elementos fundamentales de efectos visuales creados por computadora en etapa de Post Procesamiento.
- Analizar costos en relación a beneficios y gastos para el desarrollo de software similar en la industria.
- Demostrar la aplicación de los conocimientos adquiridos durante el curso en función de la implementación digital de efectos.

Alcances del proyecto.

El proyecto abarca el desarrollo de un videojuego desde cero, planteando una ideología referenciada de la industria; Poniendo sobre la mesa una idea innovadora y asegurada; revisando elementos del mercado del entretenimiento de videojuegos; como el estilo artístico, adaptabilidad de usuario, necesidades de usuario y complejidad computacional. Con el objetivo de implementar un prototipo sólido y funcional para una plataforma de gran tamaño siendo esta la computadora personal, en el que el usuario será capaz de interactuar con el prototipo desarrollado y presentar una buena experiencia.

El desarrollo del videojuego está pensado para desarrollarse en un tiempo de 3 meses, siguiendo una metodología de trabajo constante, con la meta de cumplir avances del proyecto en tiempos específicos dados por la metodología utilizada. Debido a la naturaleza del proyecto se tiene en mente que el juego sea accesible para el público de PC en general, por medio de descargas gratis y requerimientos mínimos.

Requerimientos del proyecto

El prototipo, tiene como objetivo funcional cumplir una serie de requisitos indispensables que aseguren que sea un proyecto escalable y factible, para que su presentación al mercado sea aceptada de buena manera. Proporcionando una primera impresión favorable a las empresas que puedan estar interesadas en el

prototipo. Para esto, los requisitos mínimos que debe contemplarse a la salida se dividen en dos secciones:

Requisitos funcionales.

Juego funcional

- El programa debe iniciar de manera correcta y sin errores, de la misma forma deberá cerrarse sin errores.

Optimización de elementos gráficos

- El uso de recursos para el apartado visual debería ser lo más eficiente posible, evitando hacer uso de demasiados recursos de memoria que provocan caída de fotogramas por segundo.
- El sistema debe cargar y dibujar los elementos gráficos de manera correcta a través de la pantalla de la computadora sin problemas de resolución.
- El sistema debe aplicar los efectos visuales de manera correcta según el contexto sin que interfieran con la experiencia de usuario.

Mecánicas de juego lógicas

- El juego debe responder como el usuario lo decida, no debe presentarse errores durante la jugabilidad que afecten la experiencia de usuario, además deben ser acordes al contexto desarrollado.
- El juego debe permitir al usuario comportarse de manera libre, dándole la oportunidad de completarlo conforme el usuario lo desee y llegando a un mismo final sin errores ni bloqueos inesperados.
- El juego no debe presentar errores al momento de ejecutar acciones entre las distintas escenas del juego, siendo esta la exploración y el combate.

Requisitos no funcionales

Usabilidad.

- La interacción del juego debe ser fácil e intuitiva para los usuarios nuevos, por lo cual debe tener una sección de instrucciones y acercamiento al mundo.
- El juego deberá poder volver a jugarse las veces que sea solicitado por el usuario, teniendo un registro de persistencia y eliminación de datos.

Eficiencia

- El juego debe ejecutarse de manera eficiente, respondiendo rápidamente a las acciones del usuario, en especial caso al momento de tener interacciones con entradas del usuario; entre las escenas de mundo y batalla, no tener más de 5 segundos de retraso de acciones del usuario.
- El tiempo de las acciones debe ser congruente, sin demoras de más de 5 segundos en las cargas de mundo y entre escenas.

Portabilidad

- El sistema debe ser compatible con distintas versiones del sistema operativo Windows, siendo este el sistema predominante del mercado.
- El juego puede ejecutarse en distintas resoluciones de pantalla
- Los requisitos de Hardware y software del juego deber ser en la medida de lo posible, mínimos, teniendo como referencia los requisitos de los juegos en el mercado en la actualidad.

Mantenibilidad.

- El código fuente del juego debe estar estructurado lo más eficiente posible, evitando fallas ocasionadas por el comportamiento o incluso por el usuario.
- Se debe asegurar una estructura modular para las diferentes funciones del juego para facilitar sus actualizaciones y corrección de errores.
- Los elementos gráficos utilizados en el desarrollo son creados para solo este fin, y deben estar organizados para escalamientos del sistema.

Rendimiento.

- El programa debe manejar de manera eficiente la información cargada del jugador en todo momento; siendo clave la persistencia de datos entre lo obtenido en la escena del mapa y al momento de entrar en combate, así como la lógica del sistema de turnos.
- Las operaciones en las animaciones y efectos VFX no deben causar retrasos significativos dentro del juego, así como en las acciones del jugador.
- Los elementos gráficos y efectos deben consumir el mínimo de memoria del sistema, sin que se pierda detalles.

Seguridad

- Las entradas del usuario deben ser validadas continuamente para evitar sobre escrituras o comportamientos no deseados,
- El sistema de movimiento del jugador debe ser lo más robusto posible, para no ocasionar salidas del mapa no establecidos
- El sistema de vida del jugador debe ser conciso evitando pérdidas de valores en cada combate, afectando la jugabilidad del juego.

Restricciones.

Las limitantes que se presentaron durante el desarrollo del prototipo, están relacionadas con el alcance general establecido, siendo las siguientes:

- El idioma del juego está compuesto por un el lenguaje español e inglés, siendo una mezcla de ambos.
- Una vez iniciado el juego no es posible guardar el progreso, por lo cual al salirse en medio de una partida el juego se reiniciará por completo.
- Para entender el manejo del juego a detalle es necesario consultar el manual de usuario proporcionado.
- Debido al tiempo proporcionado y a la naturaleza del proyecto el prototipo está contemplado en ser corto, pudiendo tener pequeños errores que no afectan la experiencia de usuario.

Criterios de Aceptación

- Un correcto flujo entre la navegación del juego, siendo posible iniciarla y terminarla de forma correcta, además de navegar entre las diferentes implementaciones.
- Un correcto diseño que involucra la ambientación, efectos, y mecánicas pulidas que siguen lo estipulado en el documento de diseño de juego. GDD (Game Design Document).
- Una correcta lógica implementada que no deja elementos sin utilizar y da pie a implementaciones futuras.

Metodología de Software.

Para el desarrollo del proyecto se desea optar por una técnica de desarrollo incremental de forma que la implementación se realiza en partes de forma cíclica en cada fase del desarrollo; se van agregando nuevas funcionalidades al sistema para observar la construcción del prototipo de forma rápida y flexible, logrando observar el comportamiento final en versiones más tempranas hasta su finalización. Utilizando ciclos cortos y rápidos con el propósito de obtener pequeñas funcionalidades, para apoyarnos de esta metodología se planea usar la técnica de SCRUM, donde se aplican “Sprints” (pequeños objetivos a corto plazo) en cada semana o dos semanas hasta llegar al final del tiempo estipulado.

Entregables de Fase.

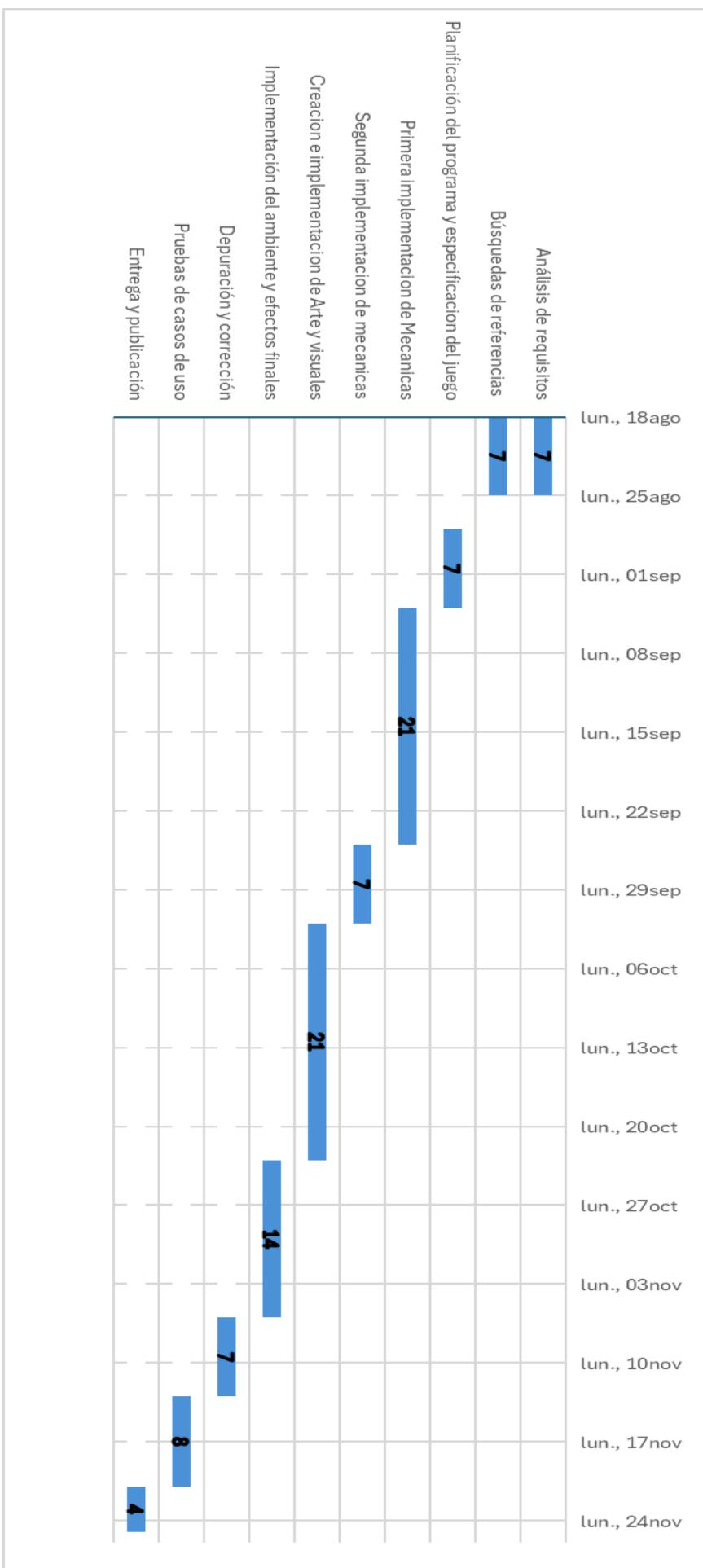
- Hitos del proyecto: Para medir y cuantificar la eficiencia de la implementación del prototipo, todas las implementaciones planeadas con ayuda de la metodología SCRUM se organizan por medio de marcas de tiempo, siendo esto los Hitos del proyecto. Cada hito representa una parte específica del desarrollo; dentro de cada hito se tienen los puntos específicos de cada Sprint en función del tiempo estipulado, siendo de 3 meses.

A continuación se presenta la tabla de hitos del proyecto:

No.	Hitos	Inicio	Fin
1	Inicio del proyecto.	18/08/2025	18/08/2025
2	Planificación del programa.	28/08/2025	28/08/2025
3	Especificaciones de elementos.	2/09/2025	4/09/2025
4	Construcción de lógicas.	4/09/2025	6/10/2025

5	Creación de visuales e implementación.	30/09/2025	21/10/2025
6	Depuración del programa.	21/10/2025	30/10/2025
7	Pruebas de casos de uso.	4/11/2025	18/11/2025
8	Publicación (fin del proyecto).	21/11/2025	25/11/2025

Con base en la tabla de hitos del proyecto, se presenta el siguiente diagrama de Gantt de actividades que comprende el proyecto para su desarrollo:



Donde según lo estipulado en los Sprints de planificación del programa y especificaciones:

Análisis de Requisitos y búsquedas de referencias (Del 18 de agosto a 22 de agosto)	<ul style="list-style-type: none">• Planteamiento del prototipo (elementos necesarios).• Planificación del sistema (Versión de Unity, tipo de renderizado).• Búsqueda de elementos de ayuda. (Referencias de juegos para inspiración)• Primera creación del GDD (Creación de la primera versión del GDD).
--	--

Planificación del programa y especificaciones del juego. (Del 28 de agosto a 4 de septiembre)	<ul style="list-style-type: none">• Creación y depuración del GDD• Creación del proyecto con los elementos necesarios(Primera implementación con elementos básicos) .• Lógicas de implementación de elementos estipulados del GDD, mecánicas generales.
--	---

Donde según lo estipulado en los Sprints de Implementación de mecánicas se planea obtener:

Primera implementación de mecánicas (Del 4 de septiembre al 25 de septiembre)	<ul style="list-style-type: none">• Lógica del mundo exterior (movimiento del jugador y enemigos).• Lógica de colisiones del mundo exterior (Colisiones entre obstáculos y enemigos).• Disparadores entre interacciones en el mundo exterior. (Lógica de transición entre combates y mundo exterior)• Lógica de patrones de diseño para manejo de datos. (Patron Singleton para persistencia de datos entre las escenas principales)• Lógica de disparadores para enemigo y jugador, así como estadísticas base.
--	--

Segunda implementación de mecánicas (Del 25 de septiembre al 2 de octubre)	<ul style="list-style-type: none">• Lógica de cambio entre escenas (Cambio de escena entre el mundo exterior y escena de combate).• Lógica de combate (Disparadores de acción del jugador y lógica de turnos, y fin de partida).• Lógica de persistencia de datos entre las escenas principales (Actualización de datos al regresar entre
---	---

	<p>escenas).</p> <ul style="list-style-type: none"> ● Lógica de implementación de sistema de inventario (recogida de elementos en el mundo exterior y persistencia de datos modificados entre las escenas de batalla) ● Lógica de implementación de elementos curativos e interacción con el inventario
--	---

Donde según lo estipulado en los Sprints de Implementación de visuales, se debe realizar:

Creacion e implementacion de arte y visuales (Del 2 de octubre al 23 de octubre)	<ul style="list-style-type: none"> ● Diseño y creación del modelado. (Escenarios, utilería, protagonista y enemigos) ● Implementación de modelos a los escenarios del juego. (Primera ambientacion con los objetos del mundo exterior y la batalla) ● Lógica y diseño de animaciones (Animaciones del enemigo y jugador en batalla y mundo exterior) ● Implementación de Animaciones (logica y controladores de animacion)
---	--

Implementación de ambiente y efectos finales (Del 23 de octubre al 6 de noviembre)	<ul style="list-style-type: none"> ● Lógica de shaders para efectos visuales dentro del juego.(Efectos de Post Procesado con volúmenes y efectos de cámara; CRT y Scanlines) ● Diseño, disposición y creación de UI UX (Interfaz de usuario para menús e interacción dentro del juego) ● Lógicas de iluminación. (Puntos de Luz y Skybox) ● Análisis de visuales. (Lógica y correcciones a la interacción con la interfaz con las mecánicas implementadas).
---	---

Con lo estipulado en la última sección de pruebas de uso y depuración se tiene:

Depuración y corrección Errores (Del 6 de noviembre al 13 de noviembre)	<ul style="list-style-type: none"> ● Lógica de Menús y flujo general (Arte y logica entre las diferentes escenas y conexión) ● Análisis de pruebas entre las escenas (Buena persistencia de datos y corrección de errores menores) ● Ajustes de errores en mecánicas (Implementación de verificadores y eliminación de datos persistentes en reinicios) ● Ajustes Visuales (optimización de la interfaz de usuario e implementacion de sonido ambiental)
--	--

Pruebas y casos de uso. (Del 13 de noviembre al 21 de noviembre)	<ul style="list-style-type: none"> ● Flujo general finalizado (Prueba general de uso del juego) ● Corrección de errores visuales. ● Corrección de animaciones. ● Implementación de características (Galería e instrucciones). ● Finalización de la documentación (GDD, manual, y proyecto).
---	---

Entrega y publicación (Del 21 de noviembre al 25 de noviembre)	<ul style="list-style-type: none"> ● Subida del repositorio actualizado ● Creación del ejecutable. ● Presentación del prototipo.
---	---

Nota: El marcado en amarillo denota actividad completada.

Riesgos del proyecto (Amenazas)

Durante toda esta implementación del prototipo y siguiendo las métricas que se tienen estipuladas en la metodología del Software se tuvo presente los siguientes riesgos que afectaron en buena y mala medida el desarrollo del prototipo:

- Retraso en la entrega de plazos: Se contempló la situación de que no se entreguen a tiempo el avance solicitado para el desarrollo, provocando retrasos al cronograma, siendo evidenciado en ciertos momentos del desarrollo.
- Cambios en las especificaciones del sistema: Se contempló el supuesto donde se modifiquen los elementos planteados desde el inicio del proyecto durante el desarrollo.
- Fallas de Software durante la implementación: Se contemplan errores durante el desarrollo, siendo algo inevitable en el proyecto presentado ocasionando retrasos.

Riesgos Positivos (Oportunidades)

- Mejoras en los plazos de entregas: Se presentó la posibilidad que se logren completar las tareas antes de lo previsto durante las entregas, permitiendo obtener un mejor beneficio y aprovechamiento.

Análisis de costos y plan financiero

En esta sección se analizará la viabilidad del desarrollo, colocando toda la información financiera que nos mostrará la rentabilidad del proyecto, como primera instancia analizaremos todos costos que se presentaron para el desarrollo del prototipo:

Inversiones y gastos de infraestructura y mantenimiento:

- Debido a las necesidades para el desarrollo del proyecto, es necesario el uso de licencias de software, entre estos datos se tienen licencias de software gráfico y arte para el modelado, diferentes herramientas de programación que se utilizaron en la implementación del prototipo, licencias del Motor Gráfico de Unity además el uso de assets para el anexo de este mismo. Para el análisis no se tiene contemplado el uso de licencias por un año ya que el desarrollo contempló un periodo de 3 meses y el tiempo posterior para actualizaciones y mantenimiento.
- Incluyendo servicios como el internet, y luz para el desarrollo y posterior actualización se tiene un valor aproximado mensualmente para cubrir el gasto.

Inversión	Precio Aproximado
Licencia: Unity Personal	\$ 0 MXN
Software de arte: krita y Blender y Aseprite	\$ 360 MXN (Aseprite)
Servicios: Luz	\$ 500 MXN mensuales
Servicios: Internet	\$ 460 MXN Mensuales
Herramientas de programación: VsCode	\$ 0 MXN
Elementos externos: Assets	\$ 100 MXN
Total (3 Meses)	\$ 3340 MXN

Esta primera inversión se contempla como costos fijos, ya que su comportamiento no cambia en función de que tanto es utilizado para el desarrollo del sistema.

Equipamiento, material y gastos de nóminas:

- El equipo necesario que fue utilizado en el proyecto consta de equipos de cómputo capaces de trabajar con el software requerido, cumpliendo las demandas de los programas que se utilizan para desarrollo. Teniendo como referencia los requisitos de Unity, se tiene:
 - Windows 10 versión 21H1 (versión 19043) o más reciente (X64), Windows 11 21H2 (versión 22000) o más reciente (Arm64).
 - GPUs compatibles con DX10, DX11, DX12 o Vulkan.
 - El proveedor de hardware soportaba oficialmente los controladores Para el desarrollo: el backend de scripting IL2CPP requiere Visual Studio 2019 con componente C++ Tools o posterior y Windows SDK versión 10.0.19041.0 o posterior.

Tomando en cuenta dichos requisitos se puede observar que los ordenadores actuales pueden cumplir sin problemas. Para el análisis se tomará como referencia el equipo PC personal del desarrollo con un costo de armado de 8,960 pesos mexicanos, con el IVA incluido. Redondeando a un valor de 9,000 pesos mexicanos. Como este bien no se encontraba nuevo al inicio del proyecto y ya estaba en uso desde fechas anteriores, es necesario obtener su valor de depreciación. Y así su costo indirecto.

Cálculo de estimación:

En México la vida útil de una computadora es de 5 años. Para el cálculo de la estimación se usa la siguiente fórmula:

$$\text{Valor Residual} = \text{Valor inicial} / 5 = 9000 / 5 = \$1800 \text{ MXN}$$

$$\text{Depreciación} = \text{Valor inicial} - \text{Valor Residual} / 5 = 9000 - 1800 / 5 = 1440 \$ \text{ MXN}$$

Al ser el primer año de la PC, el equipo tiene un valor de: \$7,560 MXN, siendo este un valor depreciado y contemplado como costo fijo.

- Para el gasto de nóminas, no se tiene un consenso de los empleados para el trabajo del proyecto, el proyecto en su totalidad fue realizado por un solo integrante, ocasionando que no exista una primera inversión y costos de suministro a los empleados del proyecto. Sin embargo, se debe considerar un costo por cada actividad durante del desarrollo de este sistema, aunque estas no tengan un costo activo.

Actividad Cronograma	Costo
Análisis de requisitos	\$ 0 MXN
Búsquedas de referencias	\$ 0 MXN
Planificación del programa y especificación del juego	\$ 0 MXN
Primera implementación de Mecánicas	\$ 400 MXN
Segunda implementación de mecánicas	\$ 400 MXN
Creacion e implementacion de Arte y visuales	\$ 700 MXN
Implementación del ambiente y efectos finales	\$ 500 MXN
Depuración y corrección Errores	\$ 0 MXN
Pruebas de casos de uso	\$ 0 MXN
Entrega y publicación	\$ 0 MXN
TOTAL	\$2000 MXN

Para el cálculo de estos valores se tomó como referencia el pago de servicios y otros gastos generales para la producción, como el material. Conformados como costos variables, ya que dependen de la producción del sistema.

Precio de venta fijo

En la siguiente tabla se muestra el costo general para el desarrollo del proyecto en el tiempo estipulado.

Detalle	Primer mes	Segundo mes	Tercer mes
Licencias	\$ 360 MXN	\$ 0 MXN	\$ 0 MXN
Servicio Luz e internet	\$ 960 MXN	\$ 960 MXN	\$ 960 MXN
Depreciación equipo	\$ 7,560 MXN	\$ 7,560 MXN	\$ 7,560 MXN
Costos Fijos	\$ 8880 MXN	\$ 8520 MXN	\$ 8520 MXN
Mano de obra	\$2000 MXN	\$2000 MXN	\$2000 MXN

directa			
Cuota plataformas (Itch.io)	\$ 0 MXN	\$ 0 MXN	\$ 0 MXN
Marketing	\$ 0 MXN	\$ 0 MXN	\$460 MXN
Costos Variables	\$2000 MXN	\$2000 MXN	\$2460 MXN
TOTAL			\$ 10,980 MXN

Al sumar los dos tipos de costos para el proyecto, se tiene finalmente una cantidad de \$10,980 MXN para el costo total que se llevó a cabo para el desarrollo del juego. Para encontrar nuestro costo de venta al público se puede utilizar la técnica de agregarle un valor de margen de ganancia al proyecto.

Margen de rentabilidad

Para un margen de ganancia y rentabilidad en proyectos de esta índole; en este caso para el prototipo de un videojuego, un porcentaje de 20% a 50% es el margen común para lograr el objetivo de rentabilidad. En este caso un valor del 35% para el margen de rentabilidad se considera un valor intermedio y correcto. Así obteniendo el valor final del proyecto a los clientes:

$$\text{Precio} = \text{Costo Total} + (\text{Costo Total} * \text{Margen de rentabilidad})$$

$$\text{Precio} = 10,980 + (10,980 * 0.35) = \$14789.15 \text{ MXN}$$

Precio redondeado: \$14,800.00 MXN

Estimación de ingresos y factibilidad

Para este rubro se proyecta la venta del título teniendo como referencia juegos similares, en este caso se toma como referencia el título de Undertale, siendo similar en mecánicas, desarrollo e idea en general.

Colocando un precio de salida igual la referencia se tiene un precio de \$5 USD, convirtiéndolo a peso nacional y redondeando se tiene un precio de \$100 MXN. Asimismo según los reportes de la plataforma de Steam la cantidad de ventas en el primer año fue la increíble cifra de 791,000 copias.

Como la página de publicación elegida, no se trata de Steam sino de Itch.io una página dedicada a juegos independientes hechos por personas entusiastas del género, los costos de comisión de la plataforma no existen o son nulos y el usuario

puede elegir la tarifa de compra para el juego (respetando la tarifa mínima propuesta por el autor) al colocar la tarifa de juego en \$ 100 MXN y agregando las tarifa extra de traslación de IVA para distribución de videojuegos de 8% y el IVA del 16%. se tiene un precio fijo final de:

$$\text{Precio despues de IVA} = 100(0.08) = 108(0.16) = \$ 125.28$$

Redondeado a \$125 MXN el precio se calcula el punto de equilibrio para observar la rentabilidad del proyecto:

$$\text{Punto de equilibrio (Unidades)} = \text{Costo Total} / \text{Precio Neto por unidad}$$

$$\text{Punto de equilibrio (Unidades)} = 14,800 / 125 = 118.4 \text{ Unidades}$$

Por lo tanto, al vender el producto a **\$100 MXN, con 119 copias vendidas se llegará al punto de equilibrio**, donde el producto empezará a ser rentable. En un periodo de un año se puede considerar realista y factible.

Requisitos Finales del Prototipo.

Finalmente, realizado el prototipo con todo lo señalado anteriormente en el presente documento, tomando aspecto de optimización y accesibilidad, además durante los casos de pruebas se tiene que tener los siguientes requerimientos en Software y Hardware hacia el usuario para presentar una buena y correcta experiencia de uso, sin errores ni fallas:

Requisitos mínimos de Software y Hardware:	<ul style="list-style-type: none">● Sistema Operativo Windows 10 u 11 de 64 bits.● Tarjeta gráfica Geforce GT 1030, 2 GB. Low Profile.● Memoria RAM 8 GB, DDR4.● Espacio de almacenamiento de 3 GB. SATA SSD (recomendado)● Directx 11
--	--

Requisitos recomendados de Software y Hardware:	<ul style="list-style-type: none">● Sistema Operativo Windows 10 u 11 de 64 bits.● Tarjeta gráfica Radeon RX 6500 XT, 4 GB.● Memoria RAM 16 GB, DDR4.● Espacio de almacenamiento de 3 GB. SATA SSD (recomendado)
---	---

- | | |
|--|---|
| | <ul style="list-style-type: none">• Directx 11 o Directx 12 |
|--|---|

Conclusión del proyecto.

Analizando el desarrollo del prototipo realizado durante un periodo de 3 meses aplicando los conceptos fundamentales del implementacion de videojuegos; conceptos que incluyen diferentes apartados como el artístico, técnico o de logísticas, se ha llegado a un resultado robusto y bien implementado de lo plasmado en las diferentes etapas del documento de desarrollo de juegos sin embargo este resultado se trata de un concepto de versión temprana, con el principal objetivo de ser escalable; respondiendo a los requerimientos del mercado en la industria se puede observar y comprobar que el uso de herramientas en especial caso los motores gráficos, lo componen una gran variedad de elementos importantes que hacen que un proyecto como este, sea posible, enfocándose en los efectos de Post Procesado que se vieron durante el curso, es un parte aguas de lo que se requiere y se experimenta dentro de este ambiente en el mercado y la vida real. La proyección vista en el este proyecto nos proporciona un panorama de cómo es implementado un medio como los videojuegos, y que aspectos se deben tomar en cuenta para que sea un proyecto factible y persistente.